BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W., WASHINGTON, D.C. 20024

# COVER SHEET FOR TECHNICAL MEMORANDUM

TITLE- The Potential for Computer Use in Spaceborne Experiments

TM-69-1031-1

DATE- March 11, 1969

FILING CASE NO(S)- 730

AUTHOR(S)- D. O. Baechler

FILING SUBJECT(S)-
(ASSIGNED BY AUTHOR(S))-

ABSTRACT

Onboard computers can increase the quantity, quality and reliability of spaceborne experiments by performing checkout, sequencing, mode control and simple data compression. Estimates of the requirements for each of these functions have been made by analogy to other systems or by outlining procedures that might be used.

Computers can also provide support for more complex data compression and reduction functions. The computer requirements for these functions vary widely from experiment to experiment so estimates have been made for specific examples such as solar flare sensing, specimen selection, categorization of multispectral data and autocorrelation of magnetic field data.

A 50-experiment mission is used to illustrate total mission requirements in terms of today's technology and that of the mid-1970's. All the experiments on this sample mission could be provided with checkout, sequencing, mode control and simple data compression by using 10% of the memory and speed capability of a computer using today's technology and less than 1% of the memory and speed capability of a mid-1970's computer.

In both cases, additional functions could be performed. A reasonable set of these is outlined in terms of the specific examples for which computer requirements were estimated. The resulting mission requirement exceeds the capability of today's computers, but in terms of mid-1970's computers it uses 4% of the storage and 20% of the speed capability. It seems clear that a spaceborne computer can be used effectively in spaceborne experiments.

(NASA-CR-106926) THE POTENTIAL FOR COMPUTER USE IN SPACEBORNE EXPERIMENTS (Bellcomm, Inc.) 37 p

N79-71540

Unclas
00/08 12729

FF No (NASA CR OR TMX OR AD NUMBER) (CATEGORY)
A[redacted]



SEE REVERSE SIDE FOR DISTRIBUTION LIST

## DISTRIBUTION

### COMPLETE MEMORANDUM TO

CORRESPONDENCE FILES:

OFFICIAL FILE COPY
plus one white copy for each
additional case referenced

TECHNICAL LIBRARY (4)

NASA Headquarters

Messrs. W. O. Armstrong/MTX
R. F. Bohling/RVA
J. L. East/REI
S. W. Fordyce/MLA
T. A. Keegan/MA-2
R. F. Lovelett/MTY
E. J. Meyers/MTE
E. P. O'Rourke/MSR-1
P. D. Schrock/MLT
G. A. Vacca/REI
J. W. Wild/MTE

Marshall Space Flight Center

Mr. H. Kerner/R-COMP-RD

Goddard Space Flight Center

Mr. G. H. Ludwig/560

Electronics Research Center

Messrs. J. W. Poduska
R. M. Snow
D. Van Meter
G. Wang

### COVER SHEET ONLY TO

Bellcomm

Messrs. F. G. Allen
G. M. Anderson
J. H. Bredt
A. P. Boysen, Jr.
K. R. Carpenter
D. A. Chisholm
D. A. DeGraaf
C. L. Davis
J. P. Downs
W. W. Elam
E. M. Grenning
D. R. Hagner
P. L. Havenstein
W. G. Heffron
H. A. Helm
J. J. Hibbert
B. T. Howard
D. B. James
A. N. Kontoratos
H. S. London
D. Macchia
E. D. Marion
J. Z. Menard
J. M. Nervik
G. T. Orrok
S. L. Penn
I. M. Ross
B. E. Sabels
J. A. Schelke
R. L. Selden
M. H. Skeer
J. W. Timko
W. B. Thompson
A. R. Vernon
J. E. Volonte
R. L. Wagner
All Members Department 1031
Department 1024 File
Central Files
Library

Bellcomm, Inc.

## TABLE OF CONTENTS

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

SUBJECT: The Potential for Computer Use in Spaceborne Experiments
Case 730

DATE: March 11, 1969

FROM: D. O. Baechler

TM-69-1031-1

TECHNICAL MEMORANDUM

## I. INTRODUCTION

### Background

In the last five years, computers have advanced sufficiently so that computational capability can now be supplied to support more than just the functions of guidance, navigation, autopiloting, and simple self-testing for which computers are used in the Apollo program. Gruman and Schaenman* described some of the additional functions for which support could be provided. Among these functions are a variety of ways in which the computer could be used to aid in spaceborne experimentation, including monitoring, confidence testing, control, and processing of experiment data for transmission to earth or for display to astronauts. Using the computer for tasks like these results in reducing the amount of time the crew must spend on monotonous jobs and increasing the information/bandwidth ratio. It may make possible experiments that would not even be considered without the availability of a computer.

This memorandum discusses some of the several ways onboard computers can be used to enhance experimentation in space. Examples are used to illustrate the feasibility of using a computer and to indicate the wide range of computer requirements which might arise.

### Guidelines

This memorandum is concerned with experiments on post-Apollo Applications Program manned space flights in earth orbit or on planetary missions. Because of the necessarily vague definition of experiments and operating philosophy for these missions, it was necessary to make many assumptions; the reader's forbearance is requested.

---

*"Functional Requirements for Spaceborne Computers on Advanced Manned Missions," E. L. Gruman and P. S. Schaenman, Proc. of Spaceborne Multiprocessing Seminar, NASA Electronics Research Center, Cambridge, Massachusetts, October 31, 1966.

Estimates of computational capability will be expressed in terms of the required number of words of memory and the required number of operations per second. The number of memory words is calculated assuming that each word has 32 bits. The number of operations per second is the number of equivalent add times required. Also, estimates of operations per second assume that a machine with approximately 50 instructions is available. Two typical instruction lists are given in Tables 1a and 1b. Neither memory estimates nor speed estimates include any allotment for computer executive functions; no particular machine configuration is assumed.

The examples described in this memorandum show the feasibility of performing various levels of computation onboard the spacecraft. For some of the examples, it will be obvious that the computer support should be provided in space; for others, it may seem that the support could be provided on the ground. The final decision should be the result of a study of such factors as the dollar cost, the available communication system (maximum bandwidth, continuity, reliability and duration of spacecraft contact with earth), and length of time necessary for a message to travel to the spacecraft (ground delays and speed of light are both factors). The experiments chosen as examples seem to be good candidates for support by spaceborne computers.

None of the examples has an explicit description of man's role in the experiment. The ordinary mode for the experiments will probably be the automatic mode. Man will provide maintenance, review experiment data, adjust equipment and take advantage of unexpected events.

## II. IMPLICATIONS OF USING COMPUTERS IN EXPERIMENTATION

One of the several implications of using computers is that the demands on crew time may be reduced, perhaps to the point where the number of crewmen could be reduced. At least, the computer would perform many of the relatively routine tasks and allow the crew to perform on a higher intellectual level. The crew's psychological adjustment to working in isolated quarters with a small number of people will be easier if tasks to be done are non-routine and non-boring.

Increased flexibility should result from using computers in experimentation. After launch, for instance, it might be possible to change a sampling rate after the experimenter has had a chance to review his initial data. He might even change the method by which the computer processes his data.

Before launch, software can be changed without the concerns of weight, volume and power consumption that normally accompany the replacement of a hardware unit. Of course, software changes are not easy to make, especially if the need for extremely high reliability dictates extensive revalidation. But if experiment software can be separated from flight-critical software and if each experiment's software can be separated from the other's, a degree of validation commensurate with the experiment's importance to the mission can be performed.

Use of computers may result in relatively less hardware onboard the spacecraft. This would be particularly true as multipurpose sensors are used more and more, so that in the limit, the $n^{\underline{th}}$ experiment might consist of simply a computer program. Thus, experimenters would build fewer complex electronic packages that had to be space qualified. Since there would be less hardware, the hardware reliability would be increased. Mission preparation times would tend to be shorter since, for some of the less critical programs, the computer programming could continue closer to launch time than hardware qualification.

The extensive use of a computer in experimentation will have an influence on the design of the computer itself. The heavy reliance on the computer will be a factor in deciding between a decentralized and centralized system and will also affect other design features. For instance, one design feature might be the capability to bypass the computer system (or at least the processing part of the system) to provide raw data directly from the sensors. This feature permits some data to be obtained even if the computer system fails or if the experiment-related software proves inadequate. It is desirable because it reduces the risk involved in relying solely on the computer and it makes some of the less critical software packages expendable and therefore less threatening to software development schedules.

Another design feature that may be incorporated to help alleviate software management problems is a memory large enough so that the software for each experiment can be assigned a section of memory and written independently from the software for other experiments. Even though this is quite inefficient in terms of memory use, it is justified because of the resulting simplification of software management.

Even with such design features, the increased use of computers in experimentation will result in increased software requirements which will in turn cause greater software management problems. Recent studies have contributed to understanding and reducing (but not removing) these problems.*

There are several long-term implications of computer use in experimentation. First, more experimenters may become interested in the space program when it becomes evident that their involvement will be primarily with software development, rather than with the building and space-qualifying of hardware. Second, as more experimenters become involved and learn of the computer capability that can be made available to them, more complex experiments may be proposed. Third, spaceborne processing may reduce experiment data to manageable proportions.

## III. USE OF COMPUTERS FOR CHECKOUT

The checkout of experiments is particularly important on long-term missions since some experiments may not be used until an appreciable time after the start of the mission and others may be used for the duration. The reasons for automating this procedure (speed, accuracy, precision, repeatability, avoidance of human error in complicated procedures) have been discussed in another memorandum.**

Computer requirements for checkout of experiments depend on the number of checkout test points, the type of checkout being performed and the frequency with which the checkout is performed.

To determine the checkout requirements that might occur, some assumptions are necessary. First, assume that monitoring an experiment requires looking at all electrical scientific data and, additionally, one experiment-hardware measurement for each scientific data measurement. Then the number of checkout test points will be twice the number of scientific data points. The number of electrical scientific data points for a set of experiments is shown in Figure 1. Only five of the 56 experiments, which are those once proposed for AAP-1, 1A, 2, 3 and 4, have more than 25 data points per experiment. More than half have five or fewer data points per experiment. But since this memorandum assumes highly sophisticated

---

*"Configuration Management of Computer Programs," Burt H. Liebowitz, Fourth Space Congress, Cocoa Beach, Florida, April 1967; "Configuration Management During the Definition and Acquisition Phases," AFSCM 375-1, US Air Force Systems Command, June 1, 1964; or "Configuration Management Exhibit for Computer Programs," ESD Exhibit EST-1, Air Force Systems Command, L. G. Hanscom Field, Bedford, Massachusetts.

**"Functional Requirements of Spaceborne Computers on Advanced Manned Missions," op. cit.

experiments, a high number of scientific data points per experiment, say 50, will be assumed. Then 100 checkout points per experiment will be monitored.

The checkout procedure itself will be passive and will be simply to compare a monitored value with an acceptable value and a range for that value. When the value is out of the acceptable range, an indication is given that might be setting a bit in a register or providing an output to sound an alarm.

The flow chart for such a procedure is shown in Figure 2. An instruction would be needed to (1) read the acceptable deviation, (2) read the measured value, (3) form the difference between the measured and acceptable values, (4) compare the difference with the acceptable deviation, (5) send an output, (6) update the index register that locates the acceptable value and deviation and (7) check an interrupt line to determine if checkout should continue. To allow some flexibility in the manner of performing these operations, storage for twice this number of instructions could be provided. If 14 words are allocated for each experiment, the programs can be written independently. The exact number is not too important because it is overshadowed by the 200 locations needed to store an acceptable value and acceptable deviation for each of the 100 checkout points assumed to be in the experiment. Therefore, a reasonable requirement is 210 words. This number would decrease if there are fewer checkout points and would increase if a more complex procedure is used.

The number of operations required to perform each test directly affects the checkout speed requirement. Using a ground rule that an experiment has to be checked out once each minute, performing 7 to 14 operations on each checkout test point would mean that about 12 to 24 operations per second are required. Remembering that not all instructions will be performed each time, 10 operations per test point is a reasonable estimate, resulting in a requirement of 17 operations per second. This number is very sensitive to the number of checkout points and to the time that is allocated to each checkout procedure.

## IV. USE OF COMPUTERS FOR CONTROL OF EXPERIMENTS

Control of experiments by computers can be open loop or closed loop. It can range from simple sequencing or mode control triggered by a parameter reaching a preset value through complex telescope pointing to using results of pattern recognition or spectral analysis for control decisions. Some of these are described below.

### Sequencing

Sequencing, that is, turning experiments on and off as the mission proceeds, can be a complex operation. Factors such as a restraint on peak power, the total power available and physical interference with a sensor by another sensor result in interactions between experiments and other systems.

If the complex process that considers these and other factors can be done on the ground to determine a simplified time line for use onboard then the onboard computer can be used for the simple task of turning each of the experiments on and off at predetermined times. It would provide the function performed by the sequencers that are sometimes used on unmanned satellites.

The advantages of using the onboard computer to perform even this simple function are (1) the flexibility of being able to make inflight changes to the predetermined sequence, and (2) the reduction in hardware resulting from replacing an existing sequencer with the computer.

The cost in terms of speed and memory requirements is small since the required onboard system can be very simple. For instance, a real time clock could be compared with a monitor register containing the time that the next on or off action is necessary and, perhaps, information about the type action necessary. A coincidence between the clock and the register would cause an interrupt. The proper control signal would be sent and the identifying word for the next action would be stored in the monitor register.

Each experiment would require 2 data words for each event associated with the experiment. One would always be needed to indicate the <u>time</u> that action is to be initiated, and another might be necessary to indicate the <u>type</u> of action to be taken if no bits in the time word are available for this use.

The program would require an instruction word for each of the following operations:

1. determine that the interrupt is the sequencing interrupt,
2. read the event number register (this is a memory location that is incremented by 1 each time a sequencing interrupt occurs),
3. based on the number in the event number register, read the word that pertains to this event,
4. send the appropriate control signal,
5. store in the monitor register the word indicating the time that the next event should take place,
6. increment and store the event number.

This program could be shared by all the experiments, but to be conservative and to permit each program to be written independently, each experiment will be allocated 6 instruction words and 2 data words. Doubling the number of instructions to cover any bookkeeping that might be necessary gives a total requirement of 14 words per experiment.

The real time requirements for this function depend on the time that can be permitted between two consecutive events. Assuming that there is at least 1 second between interrupts, only one event occurs within that second, and 10 of the 12 instructions are performed for each event, the real time requirement is 10 operations per second.

Mode Control

Mode control is similar to sequencing except that instead of following a predetermined set of steps, action may be initiated by outputs from other onboard sensors or by the output of the sensor whose mode is being changed.

Control of a camera offers an example of this type of control. The criterion for turning on the camera may be that the spacecraft is over land areas, or, even more simply, that there is no cloud cover. Assuming that the cloud cover criterion is used, and that there is a cloud cover sensor available onboard, the control function is fairly simple. In fact, the signal indicating cloud cover is analogous to the interrupt signal in the example of sequencing discussed above. Therefore the same estimates would apply. That is, the requirement is 14 memory words per experiment and 10 operations per second. These requirements restrict the mode control to being a simple response to a simple stimulus and does not include such things as calculation of a rate or an average. For instance, if a sensor is to be turned on when the spacecraft is over Long Island, New York, these requirements provide for doing so upon receiving a signal from the guidance and navigation system. They do not provide for position calculation.

Advantages of using a computer to perform this function include the capability to base the control decision on processed data from another sensor, and the reduction in hardware that results from using a computer rather than building a specialized component to perform the job.

Sequencing and mode control are two simple types of control that could be used with many experiments. Three other examples of control for specific experiments are described in the appendix to show the wide range of requirements that might arise from experiment control.

One of these--choosing among specimens by analyzing the gamma-ray spectra produced by exciting their nuclei--results in an estimated requirement of 20,000 words of memory and 200 operations per second. Another example, requiring 3,000 memory words and 12 operations per second, is a type of pointing control that might be encountered on a planetary mission. The third example is solar flare sensing. This requires 10,000 words of memory and 70,000 operations per second.

One of the ways that computers can be used for experiment control is to process data and provide displays to the astronauts. Experiment data might be presented as curves, histograms, comparative averages from several sensors or other forms for evaluation and decision making by the astronauts. What amount of processing is solely for displays, what unique processing needs to be done and what kinds of displays are necessary are topics worthy of separate study and will not be further discussed in this memorandum.

## IV. USE OF COMPUTERS FOR DATA COMPRESSION AND REDUCTION

The bits of experiment data telemetered to the ground during a space mission generally result, after analysis, in a smaller number of information bits. If some of the analysis can be done on board the spacecraft, it may be possible to send down more information without increasing the number of bits that are transmitted.

Photographs are one of the very biggest contributors to the data generated on board the spacecraft. The policy for handling photographs will directly influence the value and the degree of data compression and reduction methods that are applied to the data. If a few selected photos are transmitted to earth and the rest are saved and physically returned to earth, compression and reduction of other data can be worthwhile. If all or nearly all photos are transmitted to earth, then the savings realized by compression and reduction of the other data will represent a small percentage of the overall data transmitted unless these techniques can also be applied to photos. Whether they can or not depends in large degree on whether enough prior knowledge of the subject in the photo exists to permit use of efficient reduction techniques.

Keeping in mind this word of caution, data compression, variable sampling and data reduction are discussed below.

### Data Compression

Data compression is defined here as all those techniques which allow for recovery of the raw data except variable sampling, which is discussed in the next section. Other terms that have been used to describe the type of analysis discussed in this section are data compaction and redundancy reduction.

Two data compression techniques are prediction and interpolation. Both of these techniques allow omission of samples when the samples can be implied by the value of preceding or succeeding samples, along with an assumed progression of values.* Interpolators obtain several samples and determine the best value between the end points, as well as how far apart the end points should be, assuming that the points follow a curve described by a polynomial. Predictors simply assume that a value will follow a polynomial and check to see if the sampled value is within prescribed bounds. The polynomial that the values are assumed to follow can in either case be zero order (a horizontal line), first order (straight line) or higher.

A simple zero-order predictor is shown in Figure 3. Figuring on two instructions to read the current and previous data samples, three to do the comparison, two to add identification to the current sample and one each for storing the data, updating the current value and updating the index register, the total is ten. This covers the operations performed on a single parameter. An additional 10 words are allocated for deciding when to continue to another parameter and for setting up the necessary index registers. The total memory requirement, then, would be 20 words.

The number of operations performed on each incoming sample will be about 10 on the average since in many cases the comparison will result in the shorter route in Figure 3 being followed. The real time computer requirement will be directly proportional to the sampling rate, which can vary from one sample every few seconds to thousands of samples each section. A typical rate is 50 samples per second, resulting in a requirement of 500 operations per second.

Compression techniques can be much more complex than this predictor. To show the range of requirements that might arise, a ground-based system operated by Environmental Sciences Service Administration was studied. This system, described in the appendix, requires an estimated 1,150 words and 1400 operations per second to monitor each of 60 channels and attempt to find some correlation between the data.

Variable Sampling

Another way to increase the efficiency of information return is to provide for variable sampling of the data at the sensors. When a sensor is sampled at a fixed rate, as is the

*"Data Compression and Adaptive Telemetry," C. M. Kortman, Proc. 1965 Western Electronic Show and Convention, August, 1965; "Redundancy Reduction - A Practical Method of Data Compression," C. M. Kortman, Proc. IEEE, Volume 55, No. 3, March, 1967.

case described in the previous section, the rate must be high enough to accommodate the highest information rate from the sensor. If the information rate from the sensor is not always the same, the high sampling rate results in some degree of redundancy. The degree of redundancy increases as the information rate decreases. If it is possible to vary the sampling rate, the redundancy can be reduced.

A variable sampling system is described in the appendix and the conclusion is reached that instead of varying the sampling rate at the sensor, it is better to sense at a high rate and use the computer to determine the rate at which the information is retained.

Data Reduction

Data reduction, as defined here, is processing of data resulting in an irreversible change of form of the original data. This type of processing promises the greatest increases in the information/bandwidth ratio, but also represents the most drastic departure from the practice of transmitting all raw data to earth.

The reduction that is performed is entirely dependent on the particular experiment and it is therefore difficult to generalize the computer requirements. To obtain an idea of the range of requirements that might arise, three examples of data reduction are described in the appendix.

One of these is the analysis of multispectral data to determine land use. If the analysis is done continuously, the requirement is 1,000 words of memory and 50,000 operations per second. If the analysis is only done half the time, corresponding to being over land, for instance, then the speed requirement can be reduced to 25,000 operations per second by adding a buffer that increases the memory requirement to about 1200 words. The second example described in the appendix is an autocorrelation of magnetic field measurements resulting in a requirement of 150 words of memory and 340 operations per second. The third example is the reduction of data from a plasma experiment, resulting in a requirement of 140 words of memory and 140 operations per second.

## VI. REQUIREMENTS FOR AN ADVANCED MISSION

What will be the experiments package on an advanced mission? What experiment functions will the computer perform? These questions are not easily answered. An experiments package evolves over a long period of time and is seldom well defined in early planning stages. Although there are no details of the specific experiments, a ballpark guess at the number of experiments producing analog or digital data is about 50. This is the approximate complexity of the experiment packages of all the AAP missions lumped together. It is assumed that all 50 experiments are operating at the same time.

The functions that the computer might perform are such general functions as checkout, sequencing, mode control and simple data compression for all the experiments. Also, there may be particular experiments for which the computer might perform functions of the same complexity as those described in the appendix. The computer requirements for these general functions and the specific examples are summarized in Table 2.

The degree of computer support for experiments depends in part on the amount of computational capability available. Present-day aerospace computers can be obtained with 131K words of memory and capable of 300,000 operations per second.* As shown in Table 3, simple data compression, mode control, sequencing and checkout can be provided for all 50 experiments by using a little more than 10% of the storage capacity and only about 9% of the speed capability of a currently available computer. Bear in mind that no allowance was made for an executive program or buffers and that a currently available computer with a memory of 131K words would be somewhat bulky--about the size of two file cabinet drawers.

Nonetheless, it appears that with an existing computer it is feasible to provide such general functions for all experiments on a mission. If a currently available computer is dedicated for use with experiments, it appears feasible to do other functions as well. The complexity of these other functions may be similar to the examples described in the appendix. For example, if a procedure as complex as the analysis of rock spectra were added for two experiments, the resulting overall requirement would be 52,900 memory words and 27,250 operations per second. Alternatively, ten experiments might be provided with additional computer processing as complex as the ESSA system and another ten might have additional processing as complex as the intermittent land analysis example. This results in a total requirement of 36,400 memory words and 290,850 operations per second. There are of course many other combinations of functions that could be performed without exceeding the capability of the computer.

---

*"Trends in Aerospace Digital Computer Design," D. O. Baechler, <u>IEEE Computer Group News</u>, January 1969.

A spaceborne computer of the mid-1970's can be expected to have 3,000,000 memory words* and to be capable of 2,000,000 operations per second.** To perform the general functions for all experiments using a computer with this capability would use less than 1% of the storage capability and 1% of the speed capability. Table 4 shows the computer requirement for a mission on which general functions are performed for all 50 experiments and in addition many tasks of the complexity of those in the appendix are performed. Yet only 4% of the storage capacity and 20% of the speed capability is used. Despite all the assumptions in arriving at the estimates of requirements and in predicting future computer capability, it is clear that a spaceborne computer could support a variety of functions for a large number of experiments on mid-70's missions. Also, it is the author's opinion that once experimenters become fully aware of the tremendous potential for using the spaceborne computer system in experimentation, all of the computer capability that is available will quickly become used.

1031-DOB-jdc

D. O. Baechler

Attachment
Appendix
Figures 1 thru 6
Tables 1 thru 4

---

*The memory size is an extrapolation of the projections given in "Present and Future State-of-the-Art in Guidance Computer Memories," R. C. Ricci, NASA TN D-4224, NASA Electronics Research Center, Cambridge, Massachusetts, November, 1967.

**"The LSI Computers in Your Future," Electronic Design, Vol. 16, No. 5, March 1, 1968.

## APPENDIX

Several specific examples of ways computers can be used to aid experiments are described below. Computer requirements in terms of memory words and operations per second are estimated. While these examples might not appear on a specific mission, they indicate the different procedures that may be needed and the possible range of computer requirements.

### Pointing Control

One type of control is accurately pointing a telescope. In the Apollo Applications program, for instance, a telescope is pointed at the sun during certain periods. Another example of pointing control that might arise on a planetary mission is acquiring and tracking one of Mars' two satellites. In this case, a telescope normally used for other observations has to quickly and efficiently acquire the desired satellite, and then continue to track it. Even the tracking function presents a stringent requirement because of the proximity of the spacecraft to the satellites.

There are, of course, many variables that will affect the computer requirement for this example. The date of launch will affect the available light on the surface of Mars and on the surface of the satellites. The relative observation times for the two moons may depend on the findings of the initial observations. But even though this particular example has variables not discussed in detail here, it serves as a general example of an experiment in which pointing and holding is of primary importance.

The memory requirement is estimated to be 2000 words for instructions and 1000 words of data, or a total of 3000 words. This is based on a preliminary estimate of the memory needed to perform functions associated with the Apollo Telescope Mount in the Apollo Applications missions.*

Speed requirements are determined by finding out how often control signals must be given. On a slow encounter mission, the spacecraft will be moving about 4.5 km/sec with respect to Mars. The speed of the faster of the two Mars' moons, Phobos, is about 2 km/sec with respect to Mars. In the worst case, the relative velocity of the spacecraft to Phobos would be 6.5 km/sec and in the best case, it would be 2.5 km/sec.

---

*"Spacecraft Computer Memory Requirements for AAP - Case 600-2," J. R. Birkemeier, Memorandum for File, Bellcomm, Inc., July 17, 1967.

Appendix (Contd.)

The encounter angle, though variable, will result in a minimum distance of about 30,000 km from the spacecraft to Phobos and will also result in a relative velocity that is somewhere between the extremes. If the telescope has a 1.5° field of view, the width, x, of the field of view at a distance of 30,000 km is $x = (30,000\ \text{km})(\tan 1.5°) = 783$ km. Therefore, for a relative velocity of 4.5 km/sec between Phobos and the spacecraft, it is necessary to provide control every time the spacecraft and Phobos move 783 km relative to one another, and 783 km/ 4.5 km/sec = 174 seconds would be available for the calculations. If each of the 2000 instructions is performed three times, the speed requirement would be 3x2000/174 = 36 operations per second.

Spectra Analysis

Matching an observed spectra with a library of spectra to identify what is being looked at is a procedure that might be common to many spaceborne experiments. One example of this is choosing a rock sample. Observing the spectra of gamma rays which are produced by naturally or artificially excited nuclei can yield a complex pulse height spectrum of the sample. The pulse height spectrum must then be transformed into a differential energy spectrum to provide meaningful information concerning the sample. A numerical least-square method of analysis has been developed to accomplish this,* and the complete computer program for performing it is given in "A Numerical Least-Square Method for Resolving Complex Pulse Height Spectra." This program includes a subroutine for analyzing the data from known samples to obtain a reference library of data. Then data from an unknown sample is analyzed and the library data is used to identify the unknown. This program is written in FORTRAN for the IBM 7094, and can be adapted to any similar computer. The computer for which the program was written has an available memory for 32,000 words, but only about 25,000 words are used.

The requirement on a mission can be reduced by generating the reference library of data on the ground. The entire subroutine for library calculation can then be omitted, but it is still necessary to have sufficient memory to store this data. Further reductions may be possible if the program is written in machine language.

---

*"A Numerical Least-Square Method for Resolving Complex Pulse Height Spectra," J. I. Trombka, Goddard Space Flight Center, Greenbelt, Maryland, April, 1967, X-641-67-184.

"Non-Dispersive X-Ray Emission Analysis for Lunar Surface Geochemical Exploration," J. I. Trombka, I. Adler, R. Schmadebeck, and R. Lamothe, Goddard Space Flight Center, Greenbelt, Maryland, August 1966, X-641-66-344.

Appendix (Contd.)

About 16,000 words are necessary for the library data. These will be included in the estimate even though they are fixed data words that could be stored in a fixed memory. It is estimated that 3000 words for the program and 1000 words for the data from the unknown sample will be needed, for a total of 20,000 words.

The speed requirement depends on the amount of time that can be spent analyzing each sample. Trombka's analysis requires an estimated 6000 operations (the 3000 instructions in the program are not all performed, but loops result in some being used several times). Assuming that 30 seconds can be allowed for the processing, the speed requirement is 200 operations per second.

Solar Flare Sensing

Solar flare sensing illustrates both experiment control and data reduction. Also, it is a function for which there is good justification for being done on-board rather than on the ground. On long range missions, on-board sensing is necessary since the spacecraft may see areas of the sun that are not visible from the earth, and since there may be an unacceptable communications time lag from earth to the spacecraft.

Information concerning the flares might be used to activate cameras and other sensors, increase the sampling rate of some sensors, move filters into place to provide selective data for some sensors, and warn the astronauts if a flare is of sufficient magnitude that exposure to its radiation would be a health hazard.

A method of automated solar flare sensing has been outlined by Agarwal.* The system he describes is one in which a square array of semiconductor diodes is used to sense the sun's visible light. Each element in the array corresponds to a location on the solar disk.

There are two methods of analyzing the information from the array. The elements in the array can be scanned and their intensities digitized and stored in a buffer memory in the form of a 6-bit code (64 grey levels). Each word in the buffer memory can be compared with a threshold, and, when an element is found to be of interest, further processing of adjacent elements can be done to determine the boundary of the flare. Estimates of requirements for this method are $10^6$ six-bit memory words and $2 \times 10^5$ operations per second.

---

*Automatic Spaceborne Solar Flare Detection," R. K. Agarwal, Technical Memorandum, Bellcomm, Inc., Case 103 (to be published).

Appendix (Contd.)

Another method that might be feasible for analyzing the information is to scan the array but only store the digitized information from those elements that are of interest. This requires a comparison to be done while the scanning is being done, and it requires storing identifying information with every element of interest. A 30-bit word must be stored instead of a 6-bit word. Only 5000 words of buffer memory are needed, since statistics show that rarely will more than 5000 elements be of interest.

The requirements for this second method have been determined to be 10,000 memory words and 70,000 operations per second.

Multichannel Interpolator

A monitoring system operated by Environmental Sciences Service Administration (ESSA) in Boulder, Colorado provides an example of a system in which the data retention rate is based on the processing of the data.

This system sequentially samples the data on 50 channels and also services interrupts on 10 channels. Each time a data sample is taken, its value is compared to previous values and to predetermined bounds permitted for the data on this channel. If a value is routine, very little manipulation is necessary. It is stored about every 5 seconds for subsequent readout from the computer.

If the value is nonroutine, further processing is necessary to determine the frequency with which its value should be stored. The processing may measure the rate of change of the data on the channel being sampled or the rate of change of the data on another channel which affects the one being sampled. Because of the interdependence of the channels, the number of operations performed each time a channel is sampled is variable. But an average can be obtained by considering the computer usage factor. In this system, all necessary computations are performed and the computer remains idle 70% of the time.* The average speed for an operation is 3.5 μsec, so the number of operations per second required for each channel is

$$(10^6/3.5)(0.30/60 \text{ channels}) = 1400 \text{ operations/second}$$

for each channel.

---

*Personal communication with L. David Lewis, Environmental Sciences Service Administration, Boulder, Colorado; February 14, 1968.

Appendix (Contd.)

The size of the program is to a large extent dependent on the amount of memory set aside as a buffer, and the choice of this amount is dictated by the maximum burst rate of information input desired. In the ESSA system, the program is about 8000 words.* If 1000 words per channel are added for the buffer storage, the total is 68,000 words, or about 1150 words per channel.

Variable Sampling

A variable sampling system is shown in Figure 4.** A different criterion is used for varying the sampling rate on each of the five channels. The rate at which channel 1 is sampled is based on its high frequency component. Channel 2 is sampled only when an event occurs. Detectors A, B and C determine the rate of change in the data from channels 3, 4, and 5 and send an error signal to the secondary scheduler. (The detectors may use different means of determining the rate of change.) The secondary scheduler schedules channels 3, 4 and 5 based on the error signals and connects one of them to the primary scheduler. The primary scheduler makes assignments of channels 1 and 2 and the secondary scheduler output.

The primary and secondary schedulers act as buffers between the aperiodic channel outputs and the necessarily periodic operation of the analog/digital converter and the telemetry transmitter.

The functions of the primary and secondary scheduler could be accomplished by an onboard computer, and this would be particularly desirable if the scheduling functions were very complex or if there were a large number of channels to be scheduled. A computer would be well-suited for scheduling channels that are sampled only when an "event" occurs (such as channel 2 in Figure 5) since the event detector output could be a discrete input to the computer. The other channels, those with analog outputs, would require analog/digital conversion before a computer could be used for scheduling. If an analog/

---

*Personal communication with L. David Lewis, Environmental Sciences Service Administration, Boulder, Colorado; February 14, 1968.

**The system block diagram and description is given in "A Synopsis on Data Compression," Donald R. Weber, Proc. 1965 National Telemetering Conference, April, 1965.

digital converter is necessary on most channels, a better approach may be to provide each channel with a high speed analog/digital converter and use the system described in the previous example, multichannel interpolator.

The memory and real time requirements for a variable sampling system would probably be somewhat less than even the predictor described in the previous section, since much of the decision procedure takes place in the analyzers associated with each channel. And since adding a high speed analog to digital converter to each of the analyzers would allow the use of the system described in the previous section, no estimate is made of the requirements for the variable sampling technique.

Land Use Analysis*

An example of data processing to increase the efficiency of information return is the onboard analysis of data to determine crop or land use signature of the land passed over by a satellite. The particular signature would depend on the use being made of the land; in a well-developed area the signatures might be those of corn, oats, rye, soybeans, wheat, pasture, red clover, farmstead, bare soil, or other, while in a less developed area the signatures might be those for dense forest, grass over 8 feet high, grass 4 to 8 feet high, grass under 4 feet high, bare soil, corn, pasture, homestead, and others. In any case, there might be 16 separate signatures of interest. These can be represented by 4 bits. A multiband scanner provides 8 bits of data on each of 18 channels to be used to determine the signature associated with each resolution element. Therefore, 144 bits are processed to determine the 4 bit signature for each resolution element.

The 18 channels cover a spectrum of radiation from about 0.1 to 14 microns. The minimum angular resolution, $\theta$, using the Raleigh criterion, is $\theta = 1.22 \frac{\lambda}{D}$. The poorest resolution would occur on the 14 micron channel, and assuming an object lens diameter of $D = 1/3$ meter, the poorest resolution would be $\theta = 51 x 10^{-6}$

*The ideas for this example and much of the numerical estimates were suggested from "Remote Multispectral Sensing in Agriculture," Research Bulletin No. 832, July, 1967, Laboratory for Agricultural Remote Sensing (LARS), Purdue University, Lafayette, Indiana, and from other publications of LARS. The system described in these references is experimental; the example described in this memorandum assumes that all the experimental difficulties have been overcome. It is not meant to imply that the numbers used here describe a real or proposed system.

Appendix (Contd.)

radians. Then for orbital altitude of approximately 200 miles (say, $A = 10^6$ ft) the linear resolution, L, would be $L = \theta \cdot A = 51$ ft. If a minimum element of resolution of 100 ft. x 100 ft. is chosen, then data would be sensed every 4200 μsec. Each time the data is sensed, a decision about the associated signature must be made. Considering that there are 16 possible answers and 18 channels of input, it seems reasonable to assume that the program for making this decision has 100 instructions, 900 data words and performs 200 operations each time data is sensed. The real time requirement, then, is 200 operations/4200 μsec, or approximately 50,000 operations per second. The memory requirement is 100 words for instructions and 900 words for data, or a total of 1,000 words.

These figures assume continuous real time operation. If the experiment is made to operate only over land, then the longest land mass would be 8000 miles, followed by 8000 miles of non-land. The speed requirement could be decreased to 25,000 operations by providing storage for $1/2(8000 \text{ miles} \times \frac{144 \text{ bits}}{100 \text{ miles}})$, or an additional 180 words.

It may seem that land use analysis should be restricted strictly to use over land and that, therefore, only the second of the two estimates should be considered. There are two reasons why it is desirable to retain both estimates. First, they provide a range that may allow more experiments to be fitted to this example. Second, land use is just an example of a possible application; if data was being processed to sense schools of fish or presence of surface vessels, there may be orbits during which sensing is practically continuous.

Magnetic Field Auto-Correlation Experiment

A method of auto-correlation for a particular experiment has been described in detail* and a stored program computer has been designed to help perform this and other experiments aboard the Interplanetary Monitoring Platform - F(IMP-F). A 20.48 second telemetry sequence is divided into 256 intervals of 80 millisecond duration. During each of the first 238 intervals, 8 bits of raw data are collected in the form of a 2's complement, 7-bits-plus-sign number. No data is collected during the remaining 18 intervals in the telemetry sequence. Designating the raw data numbers as $E_i$ (i=1...238), the output numbers are

---

*"Telemetry Computer Studies," E. P. Stabler, Goddard Space Flight Center, Greenbelt, Maryland, N66-23407, X-100-65-407, September 1965; "Spacecraft Computers for Scientific Information Studies," Proceedings of the IEEE, Vol. 54, No. 12, December 1966, E. P. Stabler and C. J. Creveling.

Appendix (Contd.)

$$R_{avg} = \left[\sum_{i=1}^{238} E_i\right] \Big/ 16 \quad ,$$

which is a 12 bit number related to the average value of the magnetic field and

$$R_j = f\left[\sum_{i=1}^{238} E_{i+j} \times E_i\right] \qquad j = 0 \ldots 8$$

which are nine 12 bit numbers related to the autocorrelation data. The function, f, in the expression for $R_j$ is a conversion from integer representation to a specialized form of floating point suitable for output to the telemetry.

In "Telemetry Computer Studies," Stabler lists the number of clock units necessary to perform this processing using a computer which performs an add operation in 30 clock units. Using 30 clock units/operations as an average, his estimates can be translated into operations per second. Each 20.48 seconds, 238 standard operations of 860 clock units duration are performed, and another 6850 clock units are needed for figure conversion, telemetry preparation and resetting so the speed requirement is

$$\frac{(238)(850) + 6850 \text{ clock units}}{(30 \text{ clock units/operation})(20.48 \text{ sec})} = \frac{340 \text{ operations}}{\text{second}} \quad .$$

The memory requirement estimate is approximately 100 words for data storage and about 50 words for program. The total estimate, then, is approximately 150 words.

Plasma Experiment

A statistics computer has been designed at Goddard Space Flight Center* to provide data processing for the Plasma Experiment onboard the Explorer XXXIV satellite. To illustrate that the functions of the statistics computer could be performed by a general purpose machine, a brief description of the experiment and the processing is necessary. The output of the sensors used in

---

*"The Plasma Statistics Computer Aboard Explorer XXXIV," D. H. Schaefer and J. W. Snively, Jr., Goddard Space Flight Center, NASA X-711-67-520, October 1967.

the experiment is a series of pulses, the frequency of which is proportional to the plasma intensity. The satellite is spin-stabilized and, to define the direction the sensor is pointing, each revolution is divided into 16 parts. In interplanetary space, nearly all plasma will be detected from the direction of the sun; in the transition region of space between the magnetosphere and the solar shock wave, the plasma is expected to be homogeneously distributed. These expected distributions are shown in Figure 5.* The plasma experiment is performed to determine:

1. the location in space of the boundary between the two regions described above,

2. the characteristics of the change in the azimuthal angle versus counting-rate curve as the satellite passes through the boundary,

3. the amount of flux present, and

4. the direction at which the counting rate is greatest.

Since $2^{19}$ counts may occur during a sixteenth of a revolution, the raw data generated during a revolution is

$$16 \frac{\text{divisions}}{\text{revolution}} \times 19 \frac{\text{bits}}{\text{division}} = 304 \text{ bits per revolution.}$$

If $C_i$ is the number of pulses received from the sensor during a 22.5° interval of a revolution, then a quantity, r, can be defined as:

$$r = \frac{\frac{1}{16} \sum_{i=1}^{16} C_i^2}{\left(\frac{1}{16} \sum_{i=1}^{16} C_i\right)^2}$$

When this ratio has its maximum value of 16, it is an indication that all inputs were sensed during a single 22.5° interval, yielding a curve similar to that shown for interplanetary space

*"The Plasma Statistics Computer Aboard Explorer XXXIV," D. H. Schaefer and J. W. Snively, Jr., Goddard Space Flight Center, NASA X-711-67-520, October 1967.

Appendix (Contd.)

in Figure 5. When this ratio has its minimum value of 1, it is an indication that the same number of inputs arrived during each of the 16 intervals of the revolution. The system described for use on Explorer XXXIV provides the following information for computation of the ratio, r, on the ground:

1. The total number of pulses during a revolution, $\sum_{i=1}^{16} C_i$, is telemetered using a logarithmic representation that requires 8 bits. The logarithmic counter used to determine this value consists of two binary counters and a switching matrix.

2. The sum of the squares, $\sum_{i=1}^{16} C_i^2$, is determined using a 12 stage counter, a 27 stage counter and associated control devices. Then by using an output from the logarithmic counter in conjunction with a 27 stage shift register and selection logic, the most significant 4 bits of the sum of squares counter is commutated to the telemetry system.

3. Four bits are telemetered to indicate the interval during which the maximum count occurred. These bits are obtained by counting the intervals, storing and comparing the value of the sum of squares counter with its value during the next interval, and, when a larger count occurs, transferring the interval number to an output register.

A method of using a digital computer to obtain the same data is outlined diagrammatically in Figure 6. This method uses as the computer input the output from a counter connected to the sensor. The functions of all the other devices described above are replaced by the computational capability of the computer. The nominal revolution time of the satellite is 3 seconds. Sixteen times during the revolution the counter at the sensor is read and reset, and each time the count is compared with the previous maximum count. If the current count is found to be a new maximum, it replaces the old one and the interval number is also recorded. The sum of the counts and the sum of the squares of the counts are also appropriately incremented.

After each revolution has been completed, the exact ratio of r is calculated, and the resulting closest integer from 1 to 16 is stored for subsequent transmission. The number of the interval during which the maximum occurred, $i_{max}$, is also stored for subsequent transmission.

Appendix (Contd.)

The flow chart shows that the maximum count, $C_{max}$, is converted to logarithmic notation and the log is stored for subsequent transmission. In the system described previously, the total count, C, was transmitted in logarithmic notation. Either could be used, depending on the one desired by the ground processors. The important thing is that a calculation of the $\log_2$ is necessary and that 8 bits are to be used to express the logarithm. It is assumed that $C_{max}$ is at least $2^3$ but less than $2^{19}$. The actual number, x, which is telemetered, is, for $C_{max} = 2^n$, equal to (n-3)16 and ranges from 1 to 256--this is the same method as that used on Explorer XXXIV.

An estimate of the time and memory requirements can be made by referring to the flow diagram of the process. The loop which must be done once each interval includes a read instruction, a compare, possibly two store instructions, two additions with replacement (one preceded by a multiplication), another compare, and an index increment. The loop which is done once every 16 intervals (1 revolution) includes a multiplication and two divisions to calculate r, a subroutine to calculate the $\log_2$, and some initializing instructions. Reasonable storage requirements are estimated to be 15 instructions and 5 data locations for the first loop and 100 instructions and 20 data locations for the second loop, including the log subroutine. The total requirement, then, is 140 words.

Time requirements are based on a nominal revolution time of 3 seconds. Estimating 15 operations per interval (240 operations per revolution) plus an additional 180 operations at the end of each revolution, a total of 420 operations would have to be performed each revolution. Therefore, 140 operations per second would be required.

This method of performing the experiment provides a more precise value of the ratio, r, and an equally precise value of the count as is obtained by the system designed for Explorer XXXIV. Further, it may be possible to reduce the number of bits required for transmission, depending on the actual maximum of the counts during a single interval and on the desirability of using the maximum count for an interval rather than the total count for the sixteen intervals.

TABLE I
TYPICAL INSTRUCTION LISTS
(a) INSTRUCTION LIST FOR CDC 5360*

| COMMAND CODE | DESCRIPTION | OPERATION | EXECUTION TIME* (Microseconds) |
|---|---|---|---|
| 26 | Add | A+Operand→A | 12 |
| 27 | Replace Add | A+Operand→A Operand Address | 18 |
| 36 | Subtract | A-Operand→A | 12 |
| 06 | Convert Binary to BCD, $I_1 = 0$ | A Binary→$A_{BCD}$ | 84 |
| 06 | Convert BCD to Binary, $I_1 = 1$ | $A_{BCD}$ → $A_{BIN}$ | 84 |
| 24 | Multiply Double Length | A Operand→AC | 90 |
| 25 | Multiply and Round | A Operand→A | 90 |
| 34 | Divide Magnitude | A+Operand→A | 90 |
| 22 | Clear Add Accumulator | Operand→A | 12 |
| 01 | Store Accumulator | A→Operand Address | 12 |
| 35 | Clear Add C Register | Operand→C | 12 |
| 02 | Store C Register | C→Operand Address | 12 |
| 15 | Shift Accumulator Right, $I_{10}=1$ | $A_n \rightarrow A_{n-1}$ | 6 + 3(r+k) |
| 15 | Shift Accumulator Left, $I_{11}=1$ | $A_n \rightarrow A_{n+1}$ | 6 + 3(r+k) |
| 15 | Long Right Shift, $I_{12}=1$ | $A_nC_n \rightarrow A_{n-1}\ C_{n-1}$ | 6 + 3(r+k) |
| 15 | Long Left Shift, $I_{13}=1$ | $A_nC_n \rightarrow A_{n+1}\ C_{n+1}$ | 6 + 3(r+k) |
| 15 | Scale A & Store Exponent $I_{14}=1$ | $A_n \rightarrow A_{n+1}$ until $A_{24}=A_{23}$ | 12 + 3(r+k) |
| 15 | Scale AC & Store Exponent $I_{15}=1$ | $A_nC_n \rightarrow A_{n+1}C_{n+1}$ until $A_{24} \neq A_{23}$ | 12 + 3(r+k) |
| 32 | Logical Product | A ∧ Operand→A | 12 |
| 23 | Logical Difference | A ⊻ Operand→A | 18 |
| 30 | Logical Sum | A ∨ Operand A | 18 |
| 03 | Halt | Execution Stops | 6 |
| 16 | Branch if A Zero | Jump to Operand Address if A = 0 | 12 |
| 17 | Branch if A Positive | Jump to Operand Address if $A_{24} = 0$ | 12 |
| 13 | Branch if A Negative | Jump to Operand Address if $A_{24} = 1$ | 12 |
| 12 | Branch on Overflow | Jump to Operand Address if A Overflow | 12 |
| 11 | Branch on I/O Busy | Jump to Operand Address if Q = 1 | 12 |
| 10 | Jump Unconditionally | Jump to Operand Address | 12 |
| 00 | Transfer Unconditionally | P→Operand Address<br>Jump to Operand Address +1 | 12 |
| 31 | Transfer Return | Operand→P | 12 |
| 20 | Load Index Register | Operand→Index Reg. | 18 |
| 21 | Store Index Register | Index Reg.→Operand Address | 18 |
| 33 | Replace Increment Memory | Index Reg. + 1→Index Reg. | 18 |
| 37 | Replace Increment Index | Index Reg. + Operand Address →Index Reg.<br>Skip if Index Reg. = 0 | 18 |
| 14 | Input A (Parallel), Channels 0-7<br>(Serial) | Input Data→Operand Address | 12<br>162 |
| 05 | Output (Parallel)<br>(Serial) | Operand Address →Output Channel | 12<br>162 |
| 04 | Input (Parallel), Channels 8-18<br>(Serial) | Input Data→Operand Address | 12<br>162 |

r = Shift Count, K = 1, if r odd, I = Instruction Register C = Arithmetic Register
K = 0, if r even
*Add 6 microseconds for address modification with index register

*"5360 COMPUTER INFORMATION MANUAL", CONTROL DATA CORPORATION, OCTOBER 1965.

TABLE I

TYPICAL INSTRUCTION LISTS

(b) INSTRUCTION LIST FOR IBM 4PI/EP*

| FIXED POINT INSTRUCTIONS | | | |
|---|---|---|---|
| Instruction | Format | Instruction | Format |
| Load | RR | Subtract Logical | RX |
| Load | RX | Compare | RR |
| Load Halfword | RX | Compare | RX |
| Load and Test | RR | Compare Halfword | RX |
| Load Complement | RR | Multiply | RR |
| Load Positive | RR | Multiply | RX |
| Load Negative | RR | Multiply Halfword | RX |
| Add | RR | Divide | RR |
| Add | RX | Divide | RX |
| Add Halfword | RX | Store | RX |
| Add Logical | RR | Store Halfword | RX |
| Add Logical | RX | Shift Left Single | RS |
| Subtract | RR | Shift Right Single | RS |
| Subtract | RX | Shift Left Double | RS |
| Subtract Halfword | RX | Shift Right Double | RS |
| Subtract Logical | RR | | |
| **BRANCHING INSTRUCTIONS** | | | |
| Branch on Condition | RR | Branch on Count | RX |
| Branch on Condition | RX | Branch on Index High | RS |
| Branch and Link | RR | Branch on Index Low or Equal | RS |
| Branch and Link | RX | Execute | RX |
| Branch on Count | RR | | |

| INPUT/OUTPUT AND STATUS SWITCHING INSTRUCTIONS | | | |
|---|---|---|---|
| Instruction | Format | Instruction | Format |
| Start I/O | SI | Load PSW | SI |
| Test I/O | SI | Load PSW Special | SI |
| Halt I/O | SI | Set Program Mask | RR |
| Test Channel | SI | Set System Mask | SI |
| Direct Input | SI | Supervisor Call | RR |
| Direct Output | SI | Change Priority Mask | SI |
| **LOGICAL INSTRUCTIONS** | | | |
| Compare Logical | RR | Insert Character | RX |
| Compare Logical | RX | Store Character | RX |
| Compare Logical | SI | Load Address | RX |
| AND | RR | Shift Left Single Logical | RS |
| AND | RX | Shift Right Single Logical | RS |
| OR | RR | Shift Left Double Logical | RS |
| OR | RX | Shift Right Double Logical | RS |
| Exclusive OR | RR | | |
| Exclusive OR | RX | | |
| Test Under Mask | SI | | |

*"THE IBM 4 PI MODEL EP DATA PROCESSOR", L.S. JIMERSON AND R.B. KMETZ, 1966 AERO/SPACE COMPUTER SYMPOSIUM, SANTA MONICA, CALIFORNIA. OCTOBER 1966.

TABLE 2

SUMMARY OF COMPUTER REQUIREMENTS FOR INDIVIDUAL EXPERIMENTS

| FUNCTIONS | MEMORY REQUIREMENTS (WORDS) | | | SPEED REQUIREMENTS (OPERATIONS/SEC) |
|---|---|---|---|---|
| | DATA | PROGRAM | TOTAL | |
| CHECKOUT | 200 | 10 | 210 | 17 |
| EXPERIMENT CONTROL | | | | |
| SEQUENCING | 2 | 12 | 14 | 10 |
| MODE CONTROL | 2 | 12 | 14 | 10 |
| ROCK SPECTRA ANALYSIS | 17,000 | 3,000 | 20,000 | 200 |
| POINTING CONTROL | 1,000 | 2,000 | 3,000 | 12 |
| SOLAR FLARE SENSING | 9,000 | 1,000 | 10,000 | 70,000 |
| DATA COMPRESSION | | | | |
| SIMPLE PREDICTOR | | 20 | 20 | 500 |
| SOPHISTICATED INTERPOLATOR (ESSA SYSTEM) | 1,000 | 150 | 1,150 | 1,400 |
| DATA REDUCTION | | | | |
| LAND USE ANALYSIS (CONTINUOUS) | 900 | 100 | 1,000 | 50,000 |
| LAND USE ANALYSIS (INTERMITTENT) | 1,100 | 100 | 1,200 | 25,000 |
| MAGNETIC FIELD AUTOCORRELATION EXPERIMENT | 100 | 50 | 150 | 340 |
| PLASMA EXPERIMENT | 25 | 115 | 140 | 140 |

TABLE 3

BASIC COMPUTER REQUIREMENTS FOR A SAMPLE MISSION

| BASIC FUNCTIONS | MEMORY REQUIREMENT (WORDS) | | SPEED REQUIREMENTS (OPS/SEC) | |
|---|---|---|---|---|
| | PER EXPERIMENT | FIFTY EXPERIMENTS | PER EXPERIMENT | FIFTY EXPERIMENTS |
| CHECKOUT | 210 | 10,500 | 17 | 850 |
| SEQUENCING | 14 | 700 | 10 | 500 |
| MODE CONTROL | 14 | 700 | 10 | 500 |
| PREDICTOR | 20 | 1,000 | 500 | 25,000 |
| MISSION TOTAL | — | 12,900 | — | 26,850 |
| CURRENTLY AVAILABLE | — | 131,072 | — | 300,000 |
| PERCENT USED | — | ~10% | — | ~9% |

TABLE 4

COMPUTER REQUIREMENTS FOR A 50-EXPERIMENT MISSION

| NAME OF FUNCTION | NO. OF EXPTS. | MEMORY WORDS | % OF AVAILABLE | OPERATIONS PER SECOND | % OF AVAILABLE |
|---|---|---|---|---|---|
| (PREDICTED AVAILABLE CAPABILITY) | | (3,000,000) | (100%) | (2,000,000) | (100%) |
| CHECKOUT AND CALIBRATION | 50 | 10,500 | < 1% | 850 | < 1% |
| EXPERIMENT CONTROL | | | | | |
| SEQUENCING | 50 | 700 | < 1% | 500 | < 1% |
| MODE CONTROL | 50 | 700 | < 1% | 500 | < 1% |
| ROCK SPECTRA ANALYSIS | 1 | 20,000 | ~ 1% | 200 | < 1% |
| POINTING CONTROL | 10 | 30,000 | 1% | 120 | < 1% |
| SOLAR FLARE SENSING | 1 | 10,000 | < 1% | 70,000 | 2.5% |
| DATA COMPRESSION | | | | | |
| SIMPLE PREDICTOR | 50 | 1,000 | < 1% | 25,000 | 1.25% |
| SOPHISTICATED INTERPOLATOR | 25 | 28,750 | ~ 1% | 35,000 | 1.75% |
| DATA REDUCTION | | | | | |
| LAND USE ANALYSIS (CONT) | 1 | 1,000 | < 1% | 50,000 | 2.5% |
| LAND USE ANALYSIS (INTERMIT) | 10 | 12,000 | < 1% | 250,000 | 12.5% |
| MAGNETIC FIELD AUTOCORRELATION | 15 | 2,250 | < 1% | 5,100 | < 1% |
| PLASMA EXPERIMENT | 15 | 2,100 | < 1% | 2,100 | < 1% |
| TOTAL | | 119,000 | ~ 4% | 425,350 | ~ 20% |

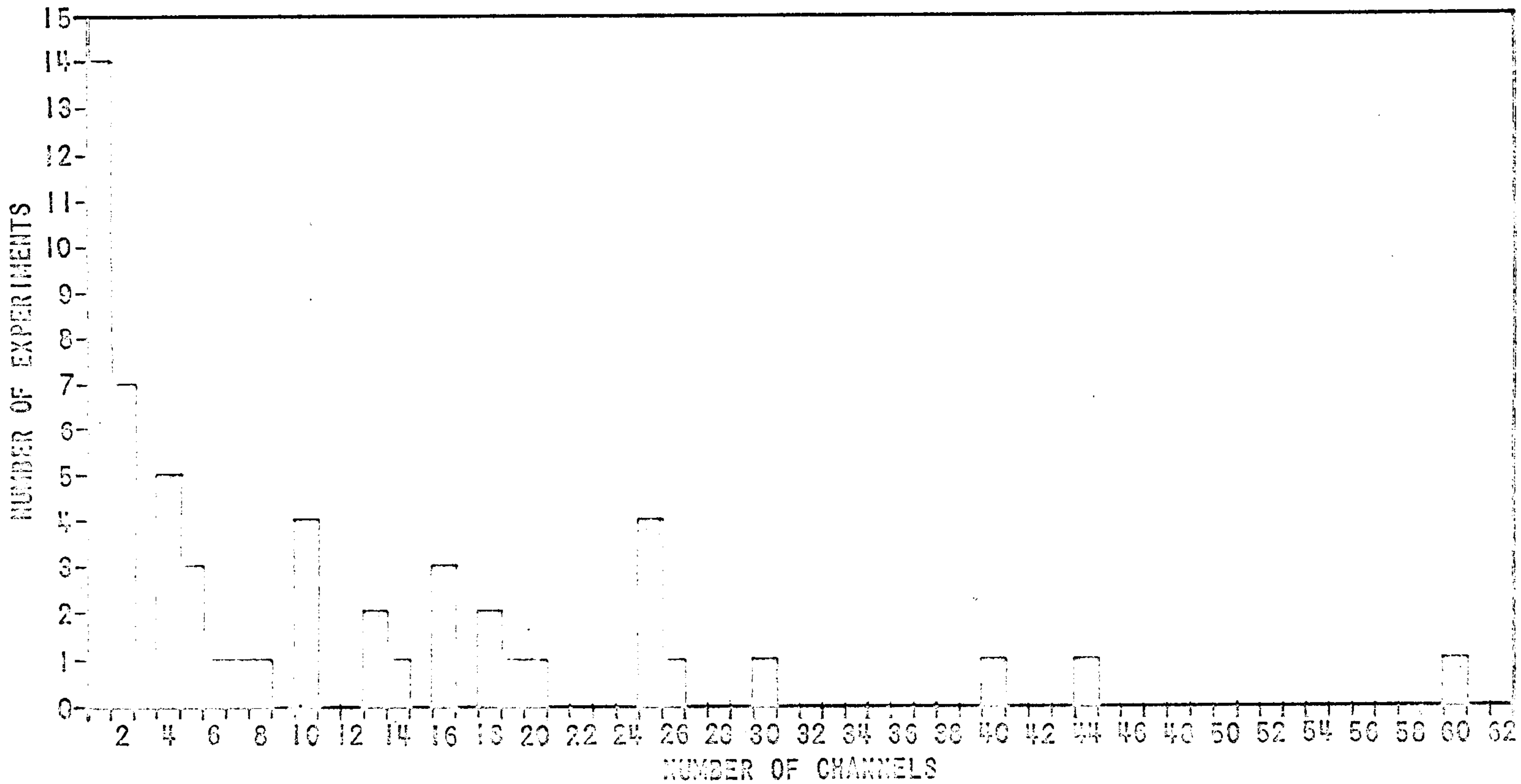



FIGURE 1 - CHANNELS FOR EXPERIMENTS ON AAP 1A, 1, 2, 3 AND 4 (AS OF 1966)

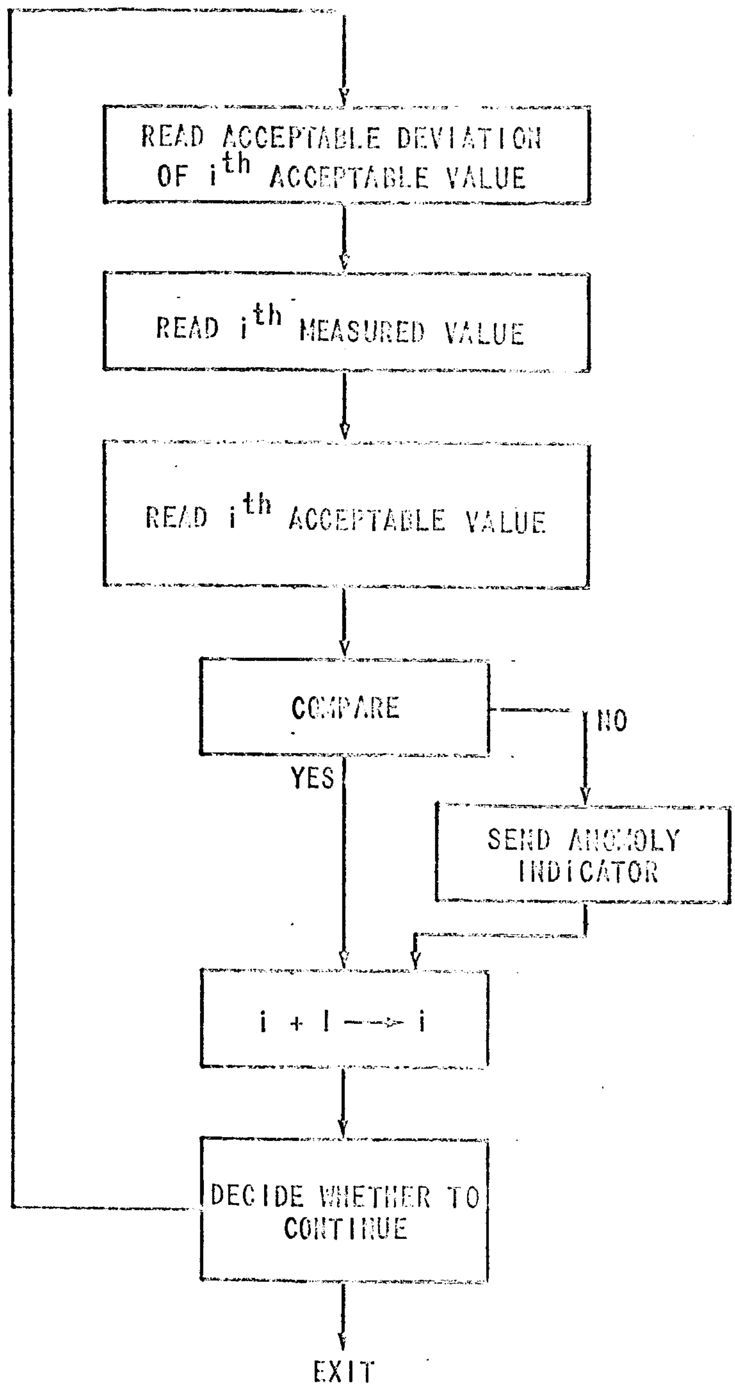



FIGURE 2 - CHECKOUT PROCEDURE

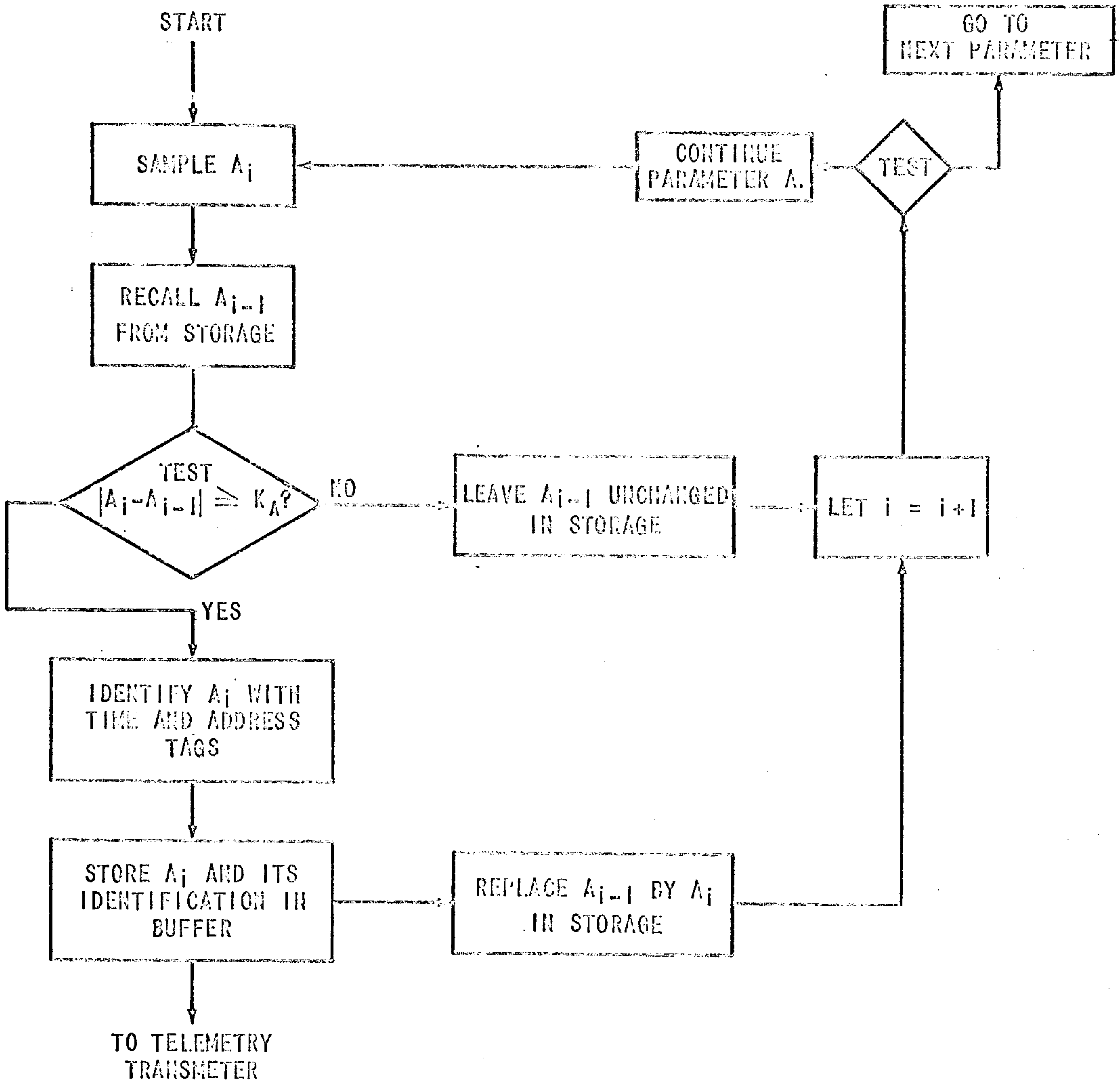



FIGURE 3 - BLOCK DIAGRAM OF STEPS IN DATA COMPRESSION USING ZERO-ORDER PREDICTOR TECHNIQUE

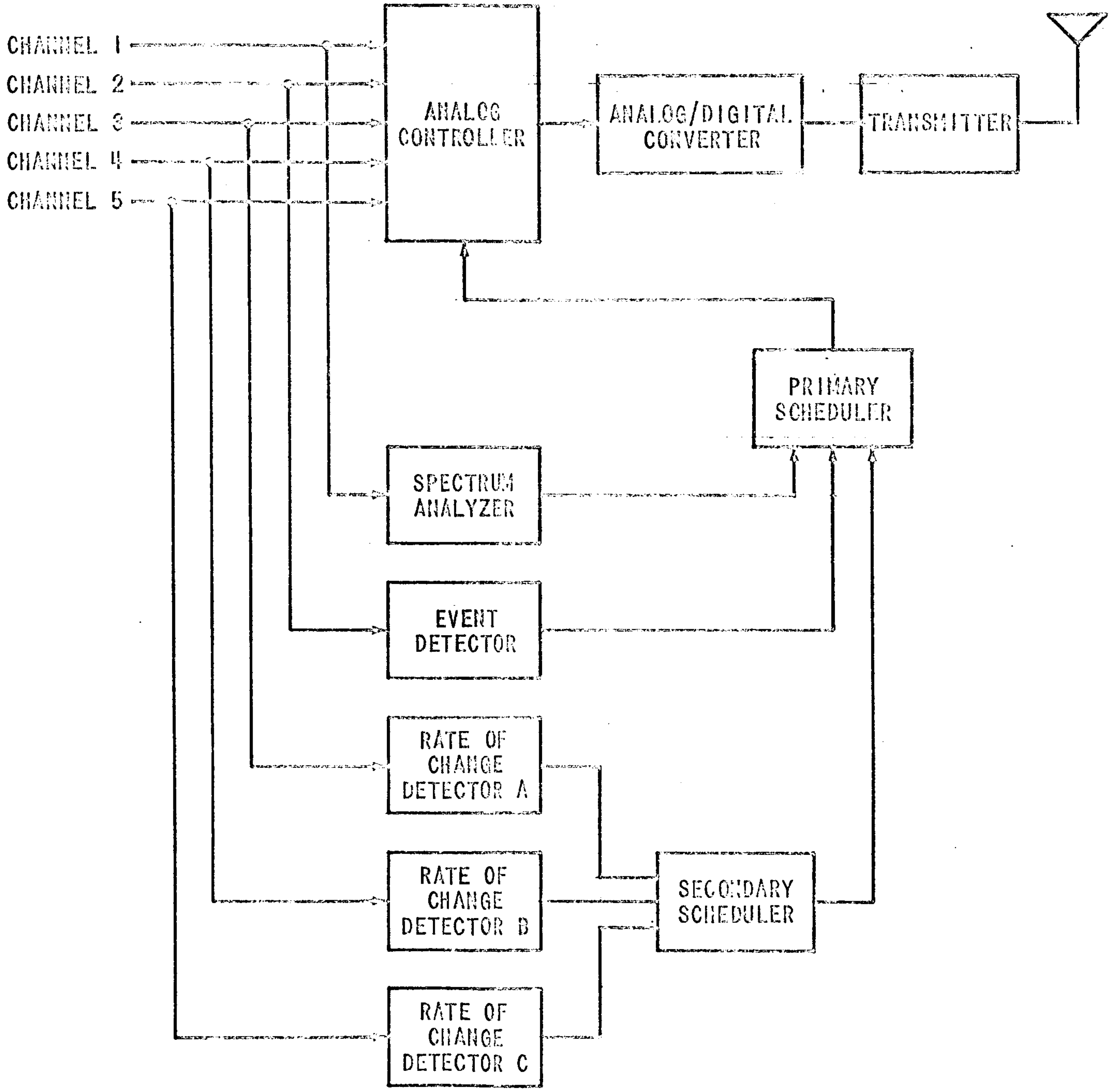



FIGURE 4 - BLOCK DIAGRAM OF VARIABLE SAMPLING SYSTEM

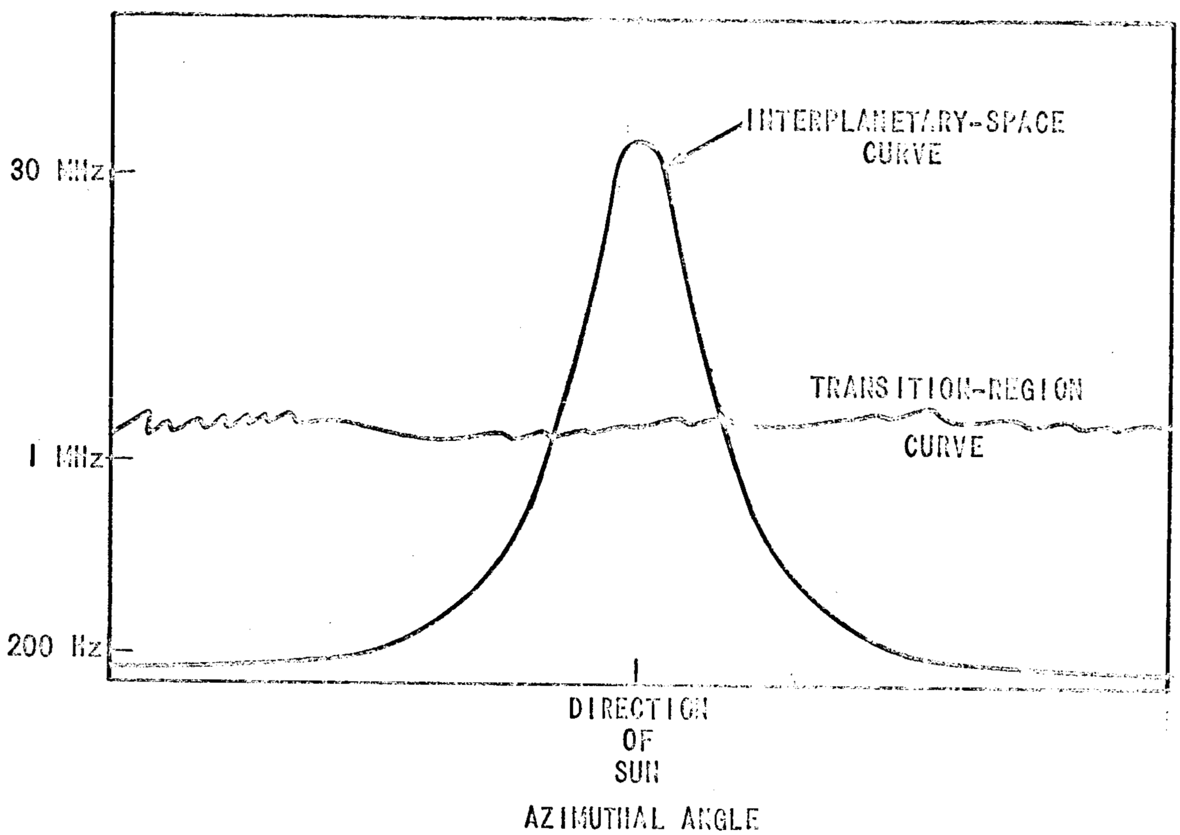
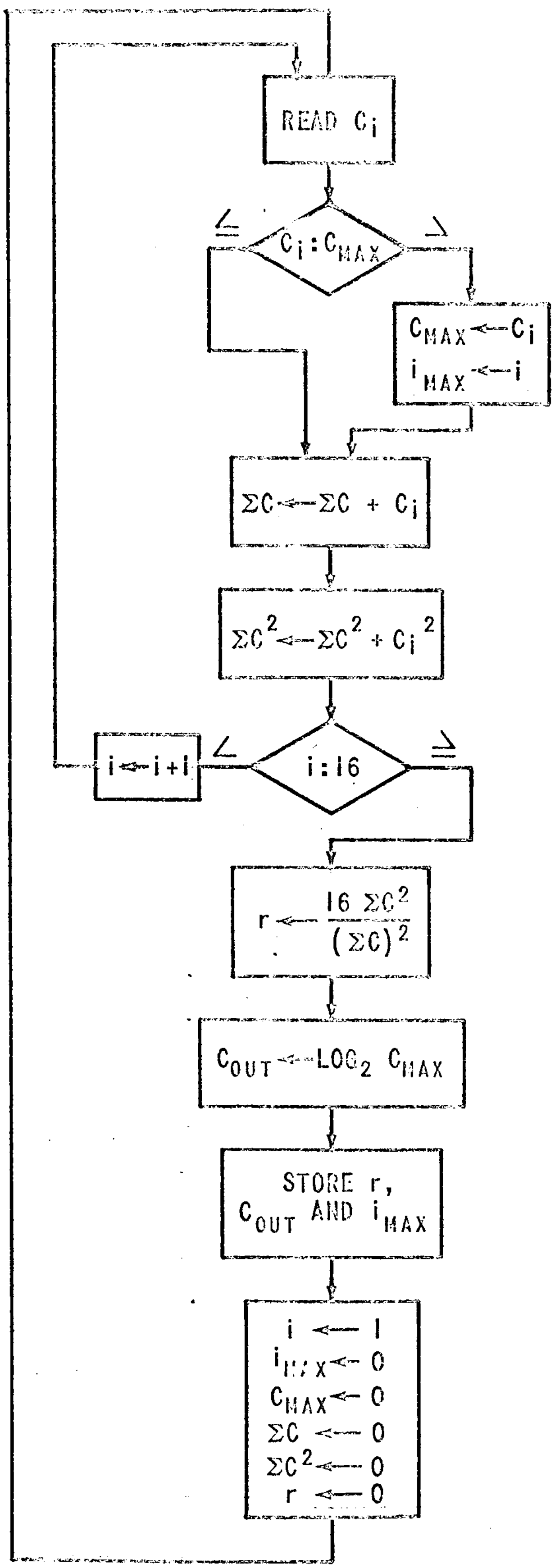



FIGURE 5 - TYPICAL OUTPUT FROM PLAZMA EXPERIMENT SENSORS




FOR EACH OF THE 16 INTERVALS, READ THE COUNTER AND RESET IT FOR THE NEXT INTERVAL. IF THIS COUNT IS THE MAXIMUM SO FAR FOR THE REVOLUTION, STORE IT AS $C_{MAX}$ AND ALSO STORE THE NUMBER OF THE INTERVAL

ADD THIS COUNT TO THE SUM OF THE COUNTS THAT HAVE OCCURRED SO FAR.

ADD THE SQUARE OF THIS COUNT TO THE SUM OF THE SQUARES OF THE COUNT SO FAR.

CONTINUE FOR THE 16 INTERVALS IN THE REVOLUTION.

AFTER EACH REVOLUTION, CALCULATE THE RATIO, r; CONVERT THE MAXIMUM COUNT TO $LOG_2$ ; AND STORE THESE QUANTITIES, AS WELL AS THE NUMBER OF THE INTERVAL DURING WHICH $C_{MAX}$ OCCURRED.

FIGURE 6 - FLOW DIAGRAM OF PLASMA EXPERIMENT DATA PROCESSING

## GLOSSARY

For those who are unfamiliar with terms used in computer literature, the following brief explanations are given for some terms as they are used in this report.

| | |
|---|---|
| Bit: | A single character in a binary number, it can have the value 0 or 1 as indicated by the absence or presence of an electrical pulse. It is analogous to digits which can have the values 0 through 9 in the decimal system. |
| Buffer: | Temporary storage used to make possible transfers between two devices whose input and output speeds are not matched. |
| Clock Time: | One clock time is equal to the period of the oscillator used to synchronize the various operations in the computer. A real time clock is a computer clock whose frequency is interpreted in terms of time outside the computer. |
| Data Points: | A location, identifiable by the computer, at which data is taken. |
| Data Word: | One of the types of words in a computer (the others are instruction words and memory words). A set of ordered bits used to represent a number. |
| Executive Function: | The coordination and supervision of various jobs in the computer. This function is performed by a computer program called the executive system or executive program. |
| Instruction: | A coded command used to tell the computer what to do. A computer has an instruction capable of making it add, another for causing a transfer, and so on. |
| Instruction Word: | One of the types of words in a computer (the others are data words and memory words). A set of ordered bits used to designate what operation the computer is to perform and what the location of the operand is. |

Glossary (Contd.)

Interrupt: An externally or internally generated signal that interrupts the current sequence of the program being performed and causes a new sequence to be performed.

Memory Word: An ordered set of bits in the computer's primary storage device. It can contain either data words or instruction words.

Operand: A word used in or resulting from an operation.

Operation: The arithmetic, logic or transfer action that the computer performs as a result of interpreting a single instruction.

Program: A list of instruction words and data that will cause the computer to operate on a problem. Usually handwritten, it is punched onto cards or tape for subsequent entry into the machine.

Register: A device for temporarily storing a single word in preparation for operating on it. It may store data, instructions, memory addresses, or any other ordered set of bits. Usually it can be loaded or emptied very quickly.

Software: All of the computer programs written for use in the computer system.